A Cloud-Based DSS Model for Cell Formation Problem

Chin-Chih Chang

Department of Information Management, Jen-Teh Junior College of Medicine, Nursing and Management

Miaoli, Taiwan

Email: chinju.chang@gmail.com

Abstract

In this study, we use the respective advantages of the tabu search (TS) and the cloud computing technologies to develop a cloud-based decision support system (DSS) for cell formation (CF) problem. To further verify the feasibility and effectiveness of the developed system, an example taken from the literature is adopted for illustrational purpose. Moreover, a set of test problems with various sizes drawn from the literature are used to test the performance of the proposed system. Corresponding results are compared to several well-known algorithms previously published. The results indicate that the proposed cloud computing CF DSS improves the best results found in the literature for 50% of the test problems. Moreover, with the assistance of our developed system, the CF practitioners in the production departments can interact with the systems without knowing the details of algorithms and can get the best machine cells and part families with maximize grouping efficacy wherever and whenever they may need it. These show that the proposed CF DSS should thus be useful to both practitioners and researchers.

Keywords

Cloud Computing; Cell Formation; Tabu Search; Decision Support System

Introduction

Cellular manufacturing (CM) is the application of group technology (GT) in manufacturing systems. GT is a manufacturing philosophy, which determines and divides the components into families and the machines into cells by taking advantage of part similarity in processing and design functions. Studies show that 30%–75% of the product cost is due to materials handling [1]. The implementation of cellular manufacturing has been reported to result in significant benefits such as reductions in material handling costs, work-in-progress inventory, throughput times and set-up times, simplified scheduling and improved quality [2].

Although CM may provide great benefits, the cellular manufacturing system (CMS) design is complex for real life problems. The foremost problem for CMS design is cell formation (CF), which groups the machines into machine cells and parts into part families. It has been known that the CF problem in CMS is one of the NP-hard combinational problems [3], as it becomes difficult to obtain optimal solutions in an acceptable amount of time, especially for large-sized problems. In this regard, many models and solution approaches have been developed to identify machine cells and part families. These approaches can be classified into three main categories [4]: (1) mathematical programming (MP) models, (2) heuristic/ meta-heuristic solution algorithms, and (3) similarity coefficient methods (SCM).

Due to their excellent performance in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic algorithm (GA), simulated annealing (SA) and tabu search (TS) have been the most successful solution approach to efficiently solve the CF problem and its variants with good results. Among the meta-heuristic algorithms, TS has been successfully used to solve many problems appeared in manufacturing system including cell formation problems [5]. Hence, we adopt it as a solver to solve the CF problem in the development of our CF cloud-based decision support system (DSS).

With the rapid development of internet and the upgrades of software and hardware technologies, the concept of cloud computing is presented. The cloud computing can provide a number of services to users through the internet whenever and wherever. Moreover, due to an increasing global competition, companies are now shifting to a geographically distributed manufacturing environment. Besides, the information flow nowadays requires reliability, efficiency and security. With the emergence of information technology, the traditional way of communication of information flows between companies and between internal parties can now be replaced by the interconnected network [6].

The Internet provides an open environment for companies to connect with their business partners as well as to serve as a medium for internal information flows [7]. Moreover, manufacturing systems have migrated to integrate with the internet to provide a remote access and control system with the characteristics of quick response and real line monitoring [8]. Thus, development of an effective computer-aided CF cloud-based DSS is necessary.

In this study, we use the respective advantages of the TS and the cloud-based technologies to develop a cloud-based DSS for CF problem. An example taken from the literature is adopted for illustrational purpose. To further verify the feasibility and effectiveness of the developed system, ten test problems with various sizes drawn from the literature are used to test the performance of our proposed CF solver. Corresponding results are compared to several well-known algorithms previously published.

The rest of the paper has been structured as follows: Section 2 describes the cell formation problem. Section 3 details the implementation of our cloud-based CF DSS. The performance of the proposed system and methodology is verified in Section 4 and the conclusion is given in Section 5.

Cell Formation Problem

Cell formation in a given 0-1 machine-part incidence matrix involves rearrangement of rows and columns of the matrix to create part families and machines cells, in which the cellular movement can be minimized and the utilization of the machines within a cell can be maximized. Two matrices shown in Figure 1 are used to illustrate the concept. **Figure 1(a)** is an initial matrix where no blocks can be observed directly. After rearrangement of rows and columns, two blocks can be obtained along the diagonal of the solution matrix in **Figure 1(b)**. For those 1's outside the diagonal blocks, they are called "exceptional elements"; while those 0's inside the diagonal blocks are called "voids".

(a)	P1	P2	P3	P4	P5
M1	0	0	0	1	0
M2	1	0	1	0	1
M3	1	1	0	1	0
M4	1	0	1	0	1
M5	0	1	0	1	0

(b)	P1	P3	P5	P2	P4
M2	1	1	1	0	0
M4	1	1	1	0	0
M1	0	0	0	0	1
M3	1	0	0	1	1
M5	0	0	0	1	1

FIGURE 1 REARRANGEMENT OF ROWS AND COLUMNS OF MATRIX TO CREATE CELLS: (A) INITIAL MATRIX AND (B) MATRIX AFTER REARRANGEMENT

There have been several measures of goodness of machine-part groups in cellular manufacturing in the literature. Two measures frequently used are the grouping efficiency [9] and the grouping efficacy [10] due to they are easy to implement. Grouping efficiency is defined as follows:

$$\eta = q\eta_1 + (1-q)\eta_2 \tag{1}$$

where η_1 is the ratio of the number of 1s in the diagonal blocks to the total number of elements in the diagonal blocks of the final matrix, η_2 is the number of 0s in the off-diagonal blocks to the total number of elements in the off-diagonal blocks of the final matrix, and q is a weight factor. Any 1s outside the diagonal blocks are called "exceptional elements," and 0s inside the diagonal blocks are called "voids."

Although grouping efficiency has been used widely, it was argued for its low discriminating capability in some cases affected by the size of the matrix. To overcome this problem, Kumar and Chandrasekharan [9] proposed another measure, the grouping efficacy, and can be defined as follows:

$$\Gamma = \frac{e - e_0}{e + e_v} \tag{2}$$

where e is the total number of 1's in the matrix, e_0 is the total number of exceptional elements, and e_v is the total number of voids. As grouping efficacy has been widely accepted in recent studies regarding CF problem, it is used as the performance measure for the proposed TS algorithm.

System Development

In this study, a cloud-based CF DSS is developed to get the best machine cells and part families with maximize grouping efficacy. The system architecture is shown in **Figure 2**. From the figure we can know that the system consists of five elements. They are the clients (i.e., users), the user interface, the web server, the CF solver and the database. All of them are linked up with the internet. We will describe them as follows:

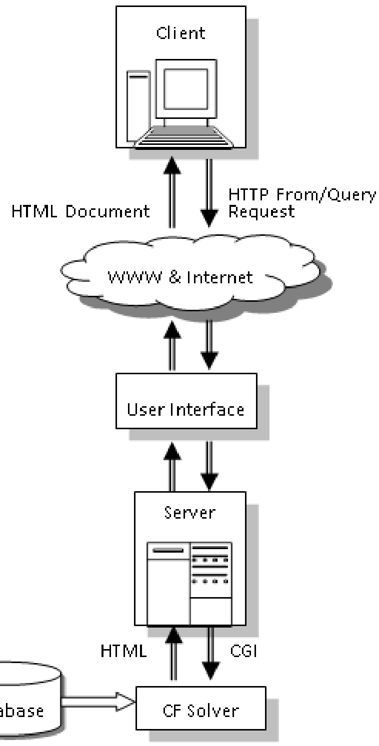


FIGURE 2 CLOUD-BASED ARCHITECTURE FOR CF DSS

Clients

web browsers are clients that connect to web servers and retrieve web pages for display. Using appropriate web browsers, such as Netscape's navigator or Microsoft's internet explorer, users can input data or view the CF results through a dynamic hypertext user interface.

User Interface

Because of the PHP is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Hence, we use PHP to making dynamic and interactive web pages for the cloud computing user interface which consists of three buttons on the top of the screen. The framework of the user interface is shown in **Figure 3** which is simple and considered to be user-friendly. We will describe them below.

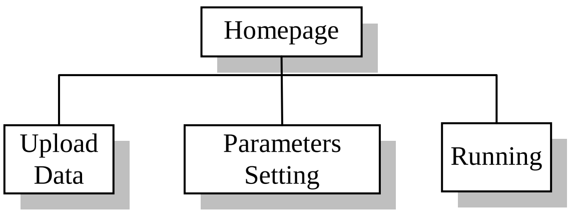


FIGURE 3 FRAMEWORK OF CLOUD COMPUTING USER INTERFACE

Web Server

The web server is a computer that serves requested web pages. The web server interacts with the individual user's web browser and accepts external Hypertext Transfer Protocol (HTTP) requests from the browser. An Application Programmer's Interface (API) is distributed, along with most of the commercially available browsers, such as Netscape's navigator or Microsoft's internet explorer. Application programs, such as PHP, ASP and JSP, can be written using these APIs to enhance the capabilities of a browser. Because of the Apache HTTP Server has been the most popular web server on the Internet since April 1996. Therefore, we use the apache server as web server in this study.

CF Solver

The CF solver was developed using Visual C++ programming languages. It consists of two stages: the initial solution construction and the improvements. The similarity coefficient-based method (SCM) is adapted in the first stage to produce good initial solutions, while the TS continuously improves and generates more effective solutions through the TS algorithm in the second stage. The proposed generic framework for the CF solver (HTSA) is shown in **Figure 4** which is actually consists of the following seven steps:

1) Initialization of computational parameters;

2) Construction of initial solution;

3) Searching of improving neighborhood solutions;

4) Update of tabu list;

5) Update of better solutions found;

6) Check of timing for directing searching toward diversified solution space by applying mutation operator;

7) Check of solution stagnancy.

Note that the first five steps are the same as the TS algorithm, while Step 6 generates new solutions with higher degree of diversification in order to increase the probability of finding the global optima, and Step 7 avoids spending too much computational efforts in order to have a balance between the computational effectiveness and efficiency.

For the CF solver, the insertion strategy is applied to produce a new neighborhood solution and the values of parameters are given below: tabu list size is equal to

7, a limit of iterations for each run is set to 1000 and the mutation probability for each gene is set at 0.8.

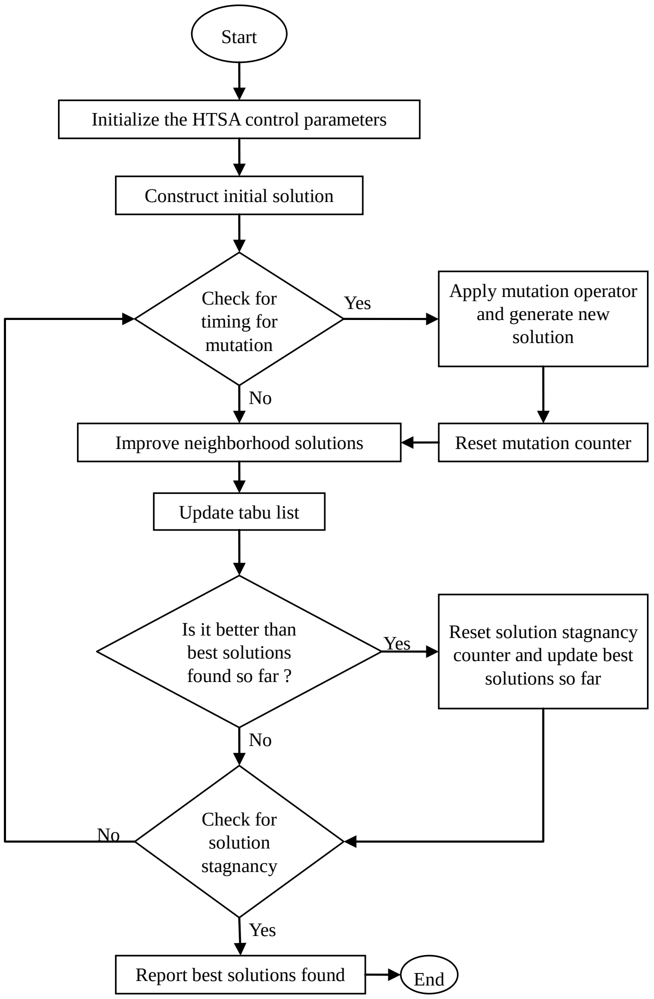


FIGURE 4 FLOWCHART OF CF SOLVER (HTSA)

Database

A database server machine may be physically different from the web server that maintains the database. Due to the MySQL is the world's most popular open source database. Hence, we use MySQL server as the database server in this study. This remote database is accessed through the Open Database Connectivity (ODBC) gateway to insert, delete or update information in the database.

Research Results

The research results consist of two parts. They are the numerical example and the comparative study. We will describe them below.

Numerical Illustration

We applied a numerical example, which was drawn from [11], to test the performance and usability of our developed system. The step-by-step procedures are described as follows:

Step 1: Press the "Upload Data" button to upload the 0-1 machine-part incidence matrix to web server, as shown in **Figure 5**, which consists of 10 machines and 15 parts.

Step 2: Set the parameters and constrains for CF solver by pressing the "Parameters Setting" button, as shown in **Figure 6**, tabu list size is equal to 7, a limit of iterations for each run is set to 1000 and the mutation probability for each gene is set at 0.8. The maximum number of machines in each cell (U_m) is limited to 4 and the minimum number of machines in each cell (L_m) is 2.

Step 3: Press the "Running" button to execute the CF solver to groups the machines into machine cells and parts into part families with maximize grouping efficacy. As shown in **Figure 7**, the CF solver only took 0.0833 seconds to get the final 0-1 machine-part incidence matrix with three cells and 92% grouping efficacy.

```
10 15
0 1 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 1 0 1 0 0 1 0 0 0 0 1 0 1
1 0 0 0 0 1 0 0 1 0 0 0 0 1 0
1 0 0 1 0 0 0 0 1 0 0 0 0 1 0
0 0 1 0 1 0 0 1 0 0 0 0 1 0 1
1 0 0 1 0 1 0 0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0 0 1 1 1 0 0 0
0 0 1 0 1 0 0 1 0 0 0 0 1 0 1
0 0 0 1 0 1 0 0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0 0 1 1 1 0 0 0
```

FIGURE 5 0-1 MACHINE-PART INCIDENCE MATRIX

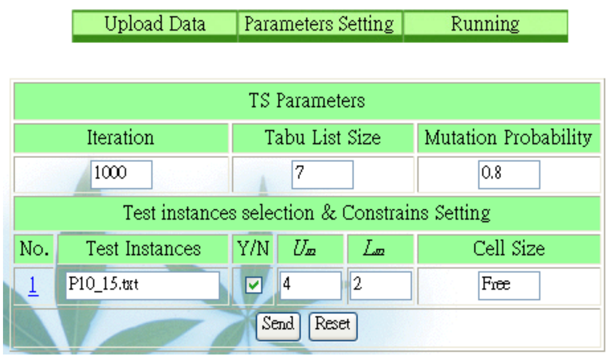
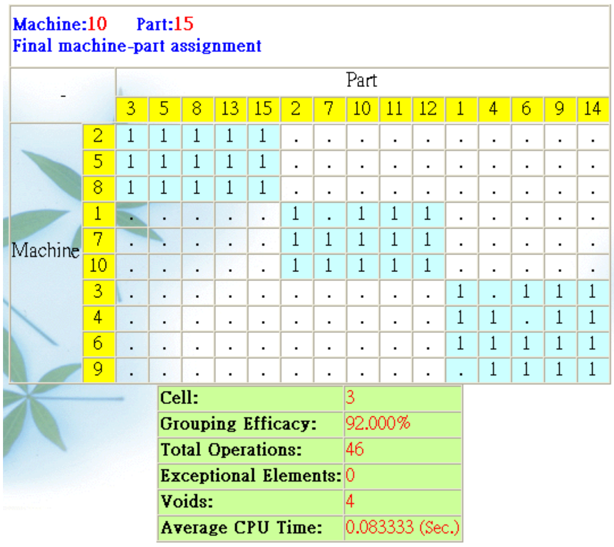


FIGURE 6 INPUT INTERFACE FOR SETTING PARAMETERS AND CONSTRAINS


Machine:10 Part:15
Final machine-part assignment

Machine \ Part	3	5	8	13	15	2	7	10	11	12	1	4	6	9	14
2	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.
5	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.
8	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	1	.	1	1	1	.	.	.	.	.
7	.	.	.	.	.	1	1	1	1	1	.	.	.	.	.
10	.	.	.	.	.	1	1	1	1	1	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.	1	.	1	1	1
4	.	.	.	.	.	.	.	.	.	.	1	1	.	1	1
6	.	.	.	.	.	.	.	.	.	.	1	1	1	1	1
9	.	.	.	.	.	.	.	.	.	.	.	1	1	1	1

Cell:	3
Grouping Efficacy:	92.000%
Total Operations:	46
Exceptional Elements:	0
Voids:	4
Average CPU Time:	0.083333 (Sec.)

FIGURE 7 OUTPUT INTERFACE FOR DISPLAYING CF RESULTS

Comparative Study

In order to evaluate the computational characteristics of our proposed CF solver with other approaches, ten test instances from the literature, as shown in **Table 1** are used. The proposed CF solver are coded in Visual C++ programming languages and implemented on an Intel(R) 1.66 GHz personal computer with 1GB RAM. **Table 2** shows the comparisons of our proposed CF solver with other approaches from the literature, that is, the EA [12], the SA [13] and the HGA [14]. The bold characters indicate the best values obtained for each test problem. It can be seen from **Table 2** that our proposed CF solver are better than or equal to those reported results. To be more specific, CF solver obtains for 5 (50%) problems values of the grouping efficacy that are equal to the best results found in EA, SA, and HGA methods and improves the values of the grouping efficacy for the rest 5 (50%) problems. It is worth to mention that our proposed CF solver is able to find the optimal solution in 5.784 seconds, illustrating the superiority of CF solver in solution efficiency.

TABLE 1 SELECTED PROBLEM FROM THE LITERATURE

No.	Source	Size
1	King [15]	16×43
2	Carrie [16]	20×35
3	Chandrasekharan and Rajagopalan [17]	24×40
4	Chandrasekharan and Rajagopalan [17]	24×40
5	Chandrasekharan and Rajagopalan [17]	24×40
6	Chandrasekharan and Rajagopalan [17]	24×40
7	Chandrasekharan and Rajagopalan [17]	24×40
8	Chandrasekharan and Rajagopalan [17]	24×40
9	Stanfel [18]	30×50
10	Stanfel [17]	30×50

TABLE 2 PERFORMANCE OF THE PROPOSED ALGORITHM COMPARED TO OTHER ALGORITHM

Test instances		Other approaches			Proposed approach		
		EA	SA	HGA	CF solver		
No.	Size	Γ (%)	Γ (%)	Γ (%)	Cell Size	Γ (%)	CPU time (s)
1	16×43	54.86	52.44	54.86	8	**56.85**	0.975
2	20×35	76.22	78.40	76.14	5	78.40	0.409
3	24×40	100.00	100.00	100.00	7	100.00	0.844
4	24×40	85.11	85.11	85.11	7	85.11	0.987
5	24×40	73.51	73.51	73.51	7	73.51	1.013
6	24×40	51.88	52.44	52.50	11	**53.29**	2.200
7	24×40	46.69	47.13	46.84	12	**48.95**	2.794
8	24×40	44.75	44.64	44.85	12	**46.26**	2.828
9	30×50	59.21	60.12	59.66	13	60.12	4.272
10	30×50	50.48	50.51	50.51	14	**50.83**	5.784

Conclusions

In this study, we have used the respective advantages of the tabu search (TS) and the cloud computing technologies to develop a cloud-based decision support system (DSS) for cell formation (CF) problem. With the assistance of CF DSS, the CF practitioners in the production departments can interact with the system without knowing the details of algorithms and can get the best machine cells and part families with maximize grouping efficacy wherever and whenever they may need it. An example taken from the literature has been adopted for illustrational purpose and a set of test problems with various sizes drawn from the literature have been used to test the performance of the CF solver. The results have indicated that the proposed algorithm improved the best results found in the literature for 50% of the test problems and the CPU times for finding the optimal solution are in 5.784 seconds. These show that our developed system should be useful to both practitioners and researchers.

REFERENCES

[1] S. Heragu, Facilities Design, MA: PWS Publishing Company, Boston (1997).

[2] U. Wemmerlov and N. Hyer, Research issues in cellular manufacturing, International Journal of Production

Research. 25, (1987).

[3] A. Kusiak, Intelligent Manufacturing Systems, NJ: Prentice Hall, Englewood Cliffs (1990).

[4] Y. Yin and K. Yasuda, Similarity coefficient methods applied to the cell formation problem: A taxonomy and review, International Journal of Production Economics. 101, (2006).

[5] S. Lozano, B. Adenso-Diaz, and L. Onieva, A one-step Tabu search algorithm for manufacturing cell design, Journal of the Operational Research Society. 50, (1999).

[6] G.Y. Tian, G. Yin, and D. Taylor, Internet-based manufacturing: a review and a new infrastructure for distributed intelligent manufacturing, Journal of Intelligent Manufacturing. 13, (2002).

[7] J. Lee, E-manufacturing systems: fundamental and tools, Robotics and Computer- Integrated Manufacturing. 9, (2003).

[8] C.S. Smith and P.K. Wright, CyberCut: a World-Wide web-based design-to-fabrication tool, Journal of Intelligent Manufacturing. 15, (1996).

[9] M.P. Chandrashekharan and R. Rajagopalan, An ideal seed non-hierarchical clustering algorithm for cellular manufacturing, International Journal of Production Research. 24, (1986).

[10] C.S. Kumar and M.P. Chandrasekharan, Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International Journal of Production Research. 28, (1990).

[11] H. M. Chan and D. A. Milner, Direct clustering algorithm for group formation in cellular manufacture, Journal of Manufacturing System. 1, (1982).

[12] J. Goncalves and M. Resende, An evolutionary algorithm for manufacturing cell formation, Computers and Industrial Engineering. 47, (2004).

[13] T.H. Wu, C.C. Chang, and S.H. Chung, A simulated annealing algorithm to manufacturing cell formation problems, Expert Systems with Applications. 34, (2008).

[14] T. Adnan, H. Iftikhar, and G. Abdu, A hybrid genetic algorithm for machine-part grouping, Computers and Industrial Engineering. 56, (2009).

[15] J.R. King, Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm, International Journal of Production Research. 18, (1980).

[16] S. Carrie, Numerical taxonomy applied to group technology and plant layout, International Journal of Production Research. 11, (1973).

[17] M.P. Chandrasekharan and R. Rajagopalan, Groupability: an analysis of the properties of binary data matrices for group technology, International Journal of Production Research. 27, (1989).

[18] L. Stanfel, Machine clustering for economic production, Engineering Costs and Production Economics. 9, (1985).